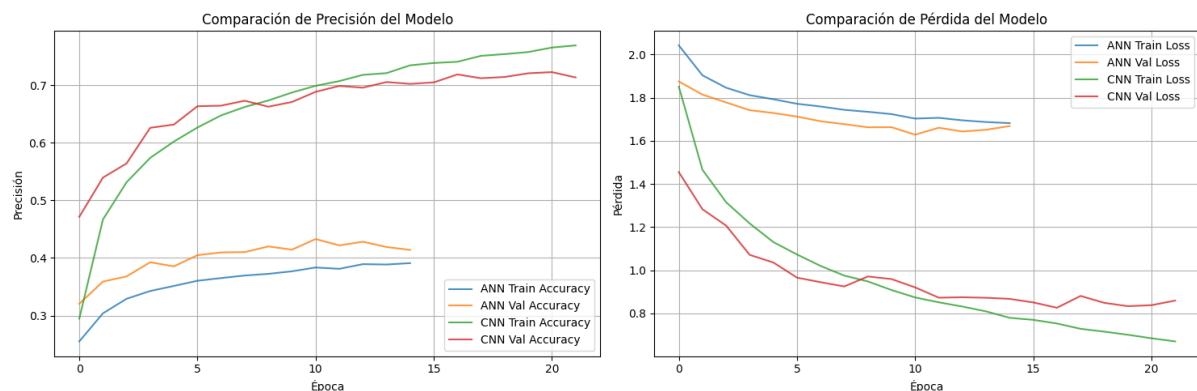


Laboratorio 4. Clasificación de imágenes con CNN

Modelo ANN y CNN

El modelo ANN tuvo un rendimiento pobre al momento de clasificar imágenes con una precisión de 0.432. A comparación de la precisión del modelo CNN que obtuvo 0.709 con una pérdida de 0.858, teniendo una mejora el CNN de 64.3% al momento de clasificar imágenes.



En esta grafica se compara la precisión del modelo ANN y CNN, en el cual las líneas de ANN celeste y naranja sube lento y se estanca alrededor de 0.28-0.42, es un patrón sub-ajuste en la que no logra relacionar correctamente la imagen. La CNN rojo y verde llega a despegar rápido y llega a 0.72 de validación, con una brecha pequeña entre entrenamiento y validación de buena generalización. La grafica de pérdida del modelo de lado derecho, la ANN reduce la pérdida muy poco (de ~2.0 a ~1.7) y se aplana pronto, confirmando la falta de capacidad/inductiva adecuada. La CNN cae fuerte ($\approx 1.85 \rightarrow \sim 0.66$ en train y ~ 0.86 en val) y luego se estabiliza; hay un pequeño “diente” en val alrededor de las épocas 8–10 (oscilación normal) que luego se corrige.

Se experimento la CNN con varios hiperparametros como el learning rate, aplicar menos drop out, y más filtros y estos fueron los resultados .

Configuración	Precisión	Pérdida
Modelo CNN original	0.709	0.858
Configuración 1: Learning rate bajo	0.706	0.870
Configuración 2: Más filtros	0.725	0.838

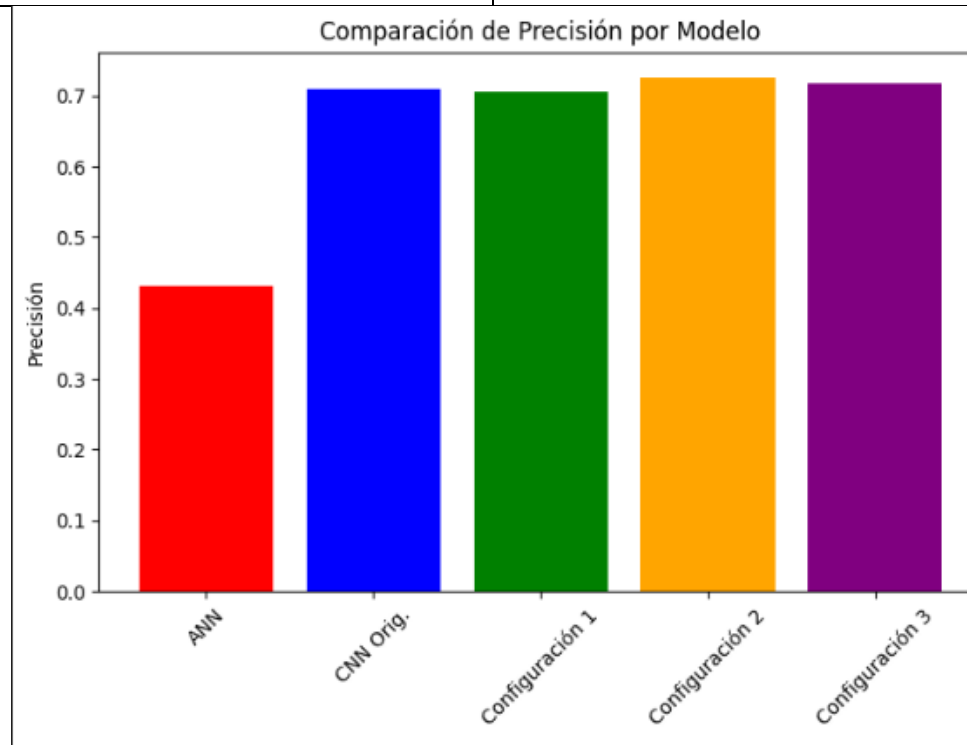
Configuración 3: Menos dropout	0.717	0.859
--------------------------------	-------	-------

La mejor configuracion fue:

Parámetro	Valor
learning_rate	0.001
dropout_rate	0.4
filters	[64, 128, 256]
dense_units	256
Precisión (test)	0.725

Comparacion final

Modelo	Precisión
ANN Original	0.432
CNN Original	0.709
Mejor CNN	0.725



Reflexión critica

¿Por qué la CNN supera (o no) al modelo ANN en este problema?

Durante el entrenamiento se ve la diferencia: la ANN “se estanca” con val_accuracy ~0.38–0.42 y val_loss > 1.7, en tanto que la CNN supera 0.60 de validación en pocas épocas y mantiene pérdidas notablemente más bajas. Es decir, la CNN aprende una representación más útil para generalizar en este conjunto de imágenes. La razón de

fondo es el sesgo inductivo correcto para visión. La ANN aplanar la imagen y rompe la estructura espacial; cada píxel se trata como una característica independiente y el modelo necesita muchísimos parámetros para “redescubrir” relaciones locales (bordes, texturas, esquinas). En cambio, la CNN usa filtros de 3x3 con pesos compartidos en toda la imagen para explorar “vecindarios” locales, esos mismos filtros se reutilizan en toda la imagen, capturando patrones que se repiten en distintas posiciones.

¿Qué papel juegan las capas de convolución y pooling?

Las capas de convolución actúan como detectores de patrones locales. Cada filtro aprende pesos que se aplican a pequeños vecindarios (receptive fields) en toda la imagen mediante compartición de pesos, produciendo mapas de características que son equivalentes a traslación. Las capas de pooling cumplen principalmente funciones de submuestreo y agregación. Al reducir la resolución espacial de los mapas de características, concentran la evidencia más relevante, disminuyen el costo computacional de las capas posteriores y aumentan la robustez a pequeñas traslaciones, deformaciones y ruido. En la práctica, el max-pooling retiene la activación más fuerte de una región, útil cuando importa la presencia de un patrón, mientras que el average-pooling promedia activaciones y produce representaciones más suavizadas.

¿Qué mejoras aplicaría? (data augmentation, arquitecturas más profundas, batch normalization).

Creo que conseguir un conjunto de datos más completo y con más referencias sería ideal. Hacer data augmentation con el conjunto que ya tenemos es lo más fácil. También considero que podemos implementar distintos optimizadores (adam, sgd, etc.) para contrastar y observar cuál es el que genera cambio de rendimiento en la CNN.